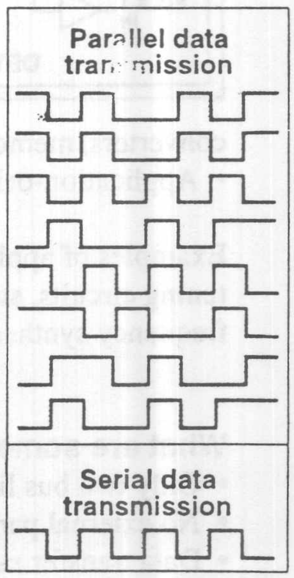
Philips' I²C (Inter-Integrated Circuit) Bus

What is a system bus?

A system bus handles communication between components (usually integrated circuits) in electronic products such as printers, computers, disk drives, calculators, televisions or telephones. The main functions of a system bus are to:

- Allow all of the system's parts to communicate with each other,
- Ensure that the parts communicate accurately,
- And when two or more parts of the system want to communicate at the same time, decide which part may communicate first. This latter problem, deciding which part can communicate first, is called arbitration.

Most microprocessor (MPU)-based systems have, as a main bus, a set of parallel wires (which are traces on a printed circuit board) in groups of eight or more (one for each bit) that connect the circuits. An entire unit of data, such as a byte, is transmitted in its component parts (bits) across all of the wires at the same time.



Most microcontroller (MCU)-based systems—found in a wide range of products including subsystems of MPU-based products, usually don't require the high speed of today's MPUs, so there is no need for a fast parallel bus. Microcontroller buses, such as the I²C bus, are generally serial buses. Individual bits of data follow each other sequentially over a wire in a serial bus.

What is I²C?

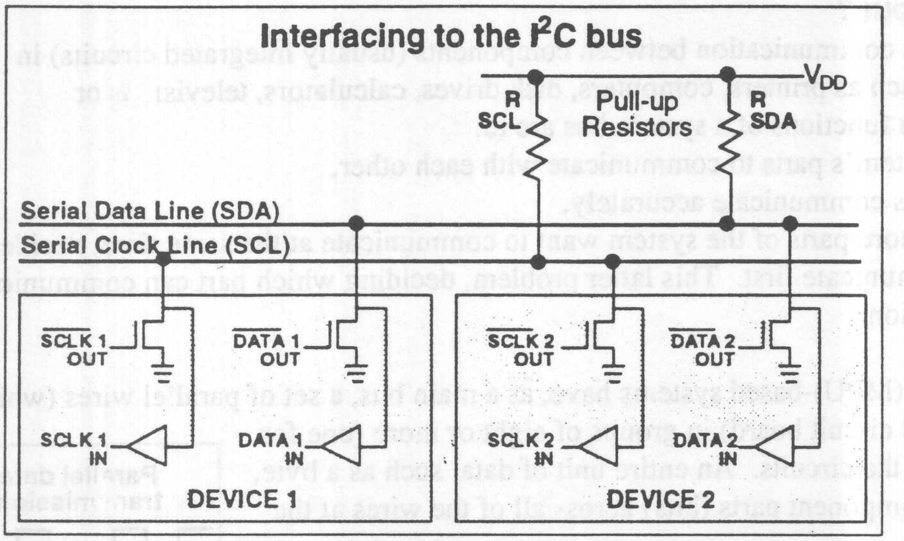
I²C is a serial bus standard developed by Philips Semiconductors that allows all circuits within a system to communicate bidirectionally with each other. Two wires are used, one carries data and the other carries timing information.

Because I²C communicates serially, the number of physical connections required between parts is reduced from eight or more to just two. This reduction cuts the number of traces on the printed circuit board (PCB) and the number of pins required on the circuits used. I²C also eliminates the extra logic circuits required to manage parallel buses.

These savings slash the complexity, size and cost of the design, thereby significantly reducing the cost of microcontroller-based products that use I²C.

How do parts of the system interface to the I²C bus?

An I²C bus interface is incorporated on-chip in every I²C integrated circuit. This means that no external components are required to tie I²C circuits to the bus. Only two pull-up resistors are required for the entire bus, one for the data line (SDA), and one for the clock (SCL) line. It is also possible to interface non-I²C circuits to an I²C bus with external logic.



What kind of circuits incorporate the I²C bus interface?

Circuits that currently incorporate the I²C bus interface include:

- A 68000-based CMOS microprocessor;
- An extensive selection of microcontrollers including a large family based on the popular 80C51 architecture;
- General purpose ICs including I/O drivers, data converters, memory, and clock calendars; and
- Application-oriented ICs for video, radio, audio, and telecom products.

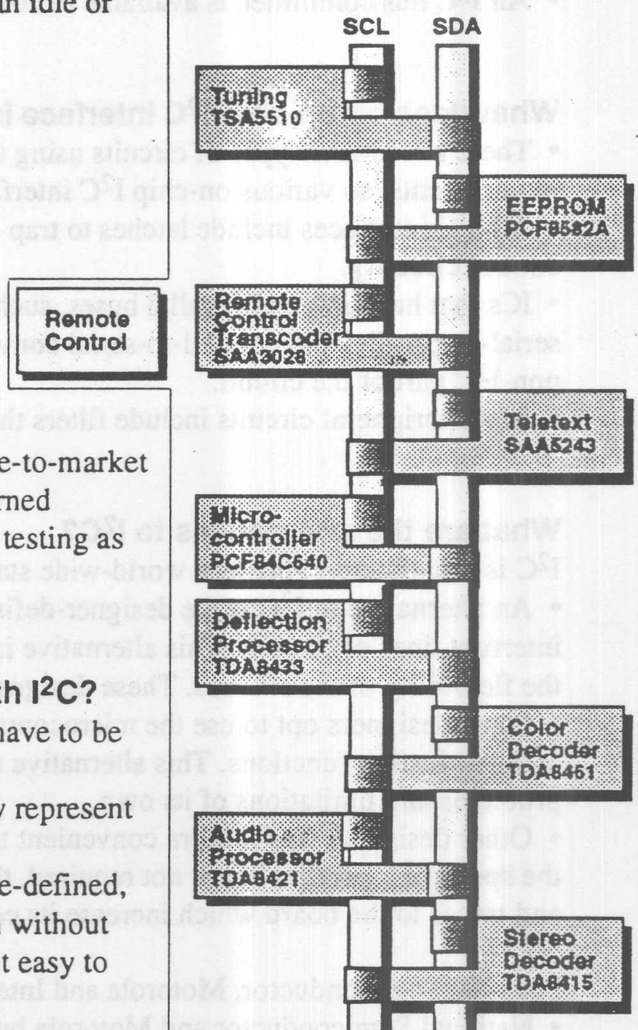
Examples of application-oriented ICs include: voice synthesizers, picture-in-picture controllers, tuning circuits, stereo components, car radio components, tone generators for telephony, and a frequency synthesizer for mobile telephones.

What are some I²C bus and circuit features?

- Only two bus lines are required—a serial data line (SDA) and a serial clock line (SCL),
- No external parts are required to connect I²C circuits to the bus,
- Data transfer rates up to 100 kilobits per second,
- Communication is not speed-dependent; fast and slow circuits can communicate over the same bus because I²C features automatic synchronization,
- Non-I²C components can be used on the bus with extra interface circuitry,
- Full multimaster capabilities—more than one master can attempt to control the bus without corrupting data,
- On-chip collision detection and arbitration,
- On-chip addressing and data-transfer protocols,
- On-chip filters on peripheral devices preserve data integrity,

• Any or all circuits on the bus can be addressed with a single address,
• Addressing is automatic; microcontrollers don't have to be slowed or stopped by interrupts to process addresses that are not theirs,
• Up to about 40 components can be connected to a single I^{2}C bus,
• Bus length may be up to 4 meters (more than 13 feet)—this length may allow connection to external devices, and
• I^{2}C bus circuits offer special features for portable equipment and battery-backed systems including:
  • Low-power options on some chips such as an idle or "sleep" mode that retains data integrity,
  • High noise immunity,
  • Different voltage supply ranges,
  • Wide operating temperature ranges.

Computer-controlled TV set.



I^{2}C bus products are easy to design and manufacture

What are some benefits of the I^{2}C bus?

• System size reduction
• Cost reduction
• Simplified design process

Because designing with the I^{2}C bus is easy, time-to-market is cut significantly, and modifications can be turned quickly. Also, the I^{2}C bus simplifies diagnostic testing as well as testing during manufacture.

How is the design process simplified with I^{2}C?

• Bus interfaces are already on-chip and don't have to be designed in,
• Functional blocks on a block diagram actually represent the layout of the board,
• I^{2}C-based systems can be completely software-defined,
• ICs can be added or removed from the system without affecting other circuits on the bus—this makes it easy to change or enhance system features,
• Malfunctions can be easily traced.

How does the I^{2}C bus work?

• Each circuit has its own bus address.
• The master generates clock signals and initiates and terminates data transfer.
• Data packets start with a 7-bit address and a single-bit that defines a request for read or write,

followed by an acknowledge bit that allows the master (initiator) to know if anyone on the bus is receiving its message.

• The data packet itself may contain an unlimited number of 8-bit bytes, each byte followed by an acknowledge bit.

• If a device on the bus can't deal with the next oncoming bit until it performs some other function—handles an interrupt, for example—it can force the master into a wait state using the clock (SCL) line. Data transfer continues after the SCL is released.

• Arbitration is handled by on-chip wired AND logic (see devices in illustration "Interfacing to the I^2C Bus" on page 2).

• An I^2C bus committee is available to coordinate the allocation of I^2C component addresses.

What does an on-chip I^2C interface include?

• There are several types of circuits using the I^2C bus, and each type has different internal requirements, so various on-chip I^2C interfaces are different from each other.

• All I^2C interfaces include latches to trap data and logic for arbitration as well as other I^2C control circuitry.

• ICs that have internal parallel buses, such as some microprocessors, for example, also include serial-to-parallel and parallel-to-serial converters to convert the data to the format used by the non-I^2C part of the circuit.

• Also, peripheral circuits include filters that ensure data integrity.

What are the alternatives to I^2C?

I^2C is rapidly emerging as a world-wide standard for microcontroller buses.

• An alternative to I^2C is the designer-defined bus that uses port pins and interrupt lines of circuits. This alternative is very design-intensive and does not offer the flexibility of the I^2C bus. These designs may also slow down bus speed.

• Some designers opt to use the microcontroller's UART to perform some inter-IC communication functions. This alternative not only uses a UART but presents design problems and limitations of its own.

• Other designers find it more convenient to work with parallel bus structures. When the speed of a parallel bus is not required, this alternative adds unnecessary circuitry and traces to the board which increase its cost and complexity.

National Semiconductor, Motorola and Intel offer alternative serial buses to I^2C.

• National Semiconductor and Motorola buses require more lines than are required by I^2C, increasing design complexity and cost.

• Intel's 9-bit bus does not allow clock synchronization—all circuits on the 9-bit bus must operate at the same speed. This is an unlikely occurrence in many designs and reduces design flexibility. In addition, some versions of the Intel bus do not offer the automatic addressing features of I^2C.

I^2C bus-related definitions

Transmitter	the device which sends data to the bus
Receiver	the device which receives data from the bus
Master	the device which initiates a transfer, generates clock signals and terminates a transfer
Slave	the device addressed by a master
Multimaster	more than one master can attempt to control the bus at the same time without corrupting the message
Arbitration	procedure to ensure that if more than one master simultaneously tries to control the bus, only one is allowed to do so and the message is not corrupted
Synchronization	procedure to synchronize the clock signals of two or more devices; these devices may operate at similar or different speeds